



- ✓ Funktioner – inbyggda
- ✓ Aggregatfunktioner
- ✓ Skalärfunktioner

Chapter 11.

Beginning SQL Server 2008 for Developers



Det finns ett mängd med funktionen du kan använd dig av. Beroende på databashanterare kan dessa variera en del. Studera därför respektive databashanteras specifika utformning av funktioner.

I MySQL se hjälpen i Query Browser.

I MSSQL se länken som leder till [SQL Server Developer Center](#)

MS SQL Server	MySQL	
ABS	ABS	Returnerar absolutbeloppet av ett tal
CEILING	CEILING	Omvandlar till heltal – uppåt
FLOOR	FLOOR	Omvandlar till heltal – nedåt
ROUND	ROUND	Avrundar tal till n decimaler
DATEADD	ADDDATE	Lägger till dagar till datum
DATEDIFF	DATEDIFF	Ex diff i antal dagar mellan två datum
MONTH	MONTH	Månadsnumret



Exempel:

```
SELECT Datum, betvillkor, DATEADD(day, betvillkor, Datum) as Fdatum  
FROM Faktura
```

Alternativ i MSSQL:

```
SELECT Datum, betvillkor, datum+betvillkor as Fdatum  
FROM Faktura;
```

Datum	Betvillkor	Fdatum
2012-03-28	30	2012-04-27
2012-04-12	20	2012-05-02
2012-04-17	15	2012-05-02

Syntax:

```
DATEADD (datepart , number, date)
```

Datepart anger typ; år, månad, dag, timmar etc. Se Books On line
Number är värdet i detta fallet antal dagar
Date är angivet datum som beräkningen går ut från



```
SELECT GETDATE() as dagens,  
YEAR(GETDATE()) as året,  
DATEPART ( YEAR , GETDATE() ) as året1,  
MONTH(GETDATE()) as månnr,  
DATEPART ( MONTH , GETDATE() ) as månnr1,  
DATENAME ( MONTH , GETDATE() ) as månnamn,  
DATEPART ( WEEK , GETDATE() ) as veckonr,  
DATEPART ( WEEKDAY , GETDATE() ) as veckodagnr,  
DATENAME ( WEEKDAY , GETDATE() ) as veckodag
```

	dagens	året	året1	månnr	månnr1	månnamn	veckonr	veckodagnr	veckodag
1	2010-02-03 08:39:01.313	2010	2010	2	2	februari	6	3	onsdag

DATENAME (datepart , date) Hämta namn med datepart
DATEPART (datepart , date) Hämta värde med datepart parameter

Se Books On Line för Datepart och Datename.



FUNKTIONER, ROUND, FLOOR, CEILING

Exempel:

```
SELECT Antal, Pris, ROUND(antal*pris,2) as Summa
FROM Artikel;
```

ROUND (number, dec) Avrundar till antal dec

Antal	Pris	Summa
10	1,75	17,50
25	1,3	32,50

Exempel:

```
SELECT Antal, Pris, FLOOR(antal*pris) as Summa
FROM Artikel;
```

FLOOR (number) Sänker till närmaste heltal

Antal	Pris	Summa
10	1,75	17
25	1,3	32

Exempel:

```
SELECT Antal, Pris, CEILING(antal*pris) as Summa
FROM Artikel;
```

CEILING (number) Höjer till närmaste heltal

Antal	Pris	Summa
10	1,75	18
25	1,3	33



SKALÄRFUNKTIONER

Skalärfunktioner arbetar mot ett värde och returnerar detta värde modifierat.

MS SQL Server	MySQL	
UPPER(k)	UCASE(k)	Returnerar strängen i VERSALER (Uppercase)
LOWER(k)	LCASE(k)	Returnerar strängen i gemener (Lowercase)
SUBSTRING(k, start, nr)	MID(k, start, nr)	Returnerar nr antal tecken från tecknet start i kolumnen k.
GETDATE()	NOW()	Returnerar aktuellt datum och tid (på servern)
ISNULL(k, nytt)	IFNULL(k, nytt)	Byter ut NULL mot nytt i kolumnen k
CAST/CONVERT	CAST	Konvertera från ett format till annat

Givetvis så finns det en hel uppsjö med ytterligare specialfunktioner beroende på databashanterare. Här är respektive databashanterares hemsida på nätet en guldgruva.

Exempel:

```
SELECT UPPER(Produktnamn) AS NAMN  
FROM Produkt;
```

NAMN
STOLEN GUNNAR
BORDET JAN
PALLEN GUN
KASSEN JARL
KVITTO



UPPER omvandlar alla tecken till versaler

Exempel:

```
SELECT UPPER(Artnamn) AS NAMN  
FROM Artikel;
```

NAMN
DVD SPELARE
CD-SKIVOR
BILDSKÄRM
TANGENTBORD
USB MINNE

SUBSTR läser av antal tecken

Exempel:

```
SELECT SUBSTRING(postnr,1,3) AS Region  
FROM Kund;
```

Region
393
394
395



CAST AND CONVERT

Att slå samman data av lika datatyper och olika datatyper

Exempel:

```
SELECT Fnamn + ' ' + Enamn as Namn
FROM Personal;
```

Fnamn	Enamn	Namn
Anna	Ek	Anna EK
Sture	Gran	Sture Gran

Exempel:

```
SELECT Avgift+Stöd as Total
FROM Medlem;
```

Avgift	Stöd	Total
100	50	150
225	30	255

Exempel:

```
SELECT Artikelnamn + ' ' + CAST(pris*antal as varchar(10)) AS
Produkt FROM Artikel;
```

Artnamn	Pris	Antal	Produkt
Stol	50	10	Stol 500
DVD	30	50	DVD 1500

Alternativ:

```
SELECT Artikelnamn + ' ' + STR(pris*antal) AS Produkt
FROM Artikel;
```




CAST AND CONVERT

CAST och CONVERT kan vi använda om vi vill ta bort visning av timmar, minuter och sekunder från datum:

Exempel:

```
SELECT Datum  
FROM Faktura;
```

Formatet på datatypen smalldatetime i SQL Server är förutom datum också hh:mm:ss

Datum
2009-01-05 00:00:00
2009-01-15 00:00:00
2009-01-25 00:00:00
2009-02-08 00:00:00
2009-02-04 00:00:00

Exempel:

```
SELECT CONVERT(char(10), datum, 120) AS Datumet  
FROM Kund;
```

Datumet
2009-01-05
2009-01-15
2009-01-25
2009-02-08
2009-02-04

Konverterar till
datatypen
char(10)

Vad ska
konverteras

Hur det ska konverteras
Se Books OnLine under
CAST AND CONVERT



AGGREGATFUNKTIONER

Aggregatfunktioner arbetar mot en grupp av värden men returnerar bara ett värde.

COUNT(kolumn)	Returnerar antalet rader exkl. NULL-poster
COUNT(*)	Returnerar antalet rader inkl. NULL-poster
AVG(kolumn)	Returnerar medelvärdet av posterna
MAX(kolumn)	Returnerar högsta värdet i kolumnen
MIN(kolumn)	Returnerar lägsta värdet i kolumnen
SUM(kolumn)	Returnerar summan av värdena i kolumnen

Beroende på databashanteraren så kan det finnas fler aggregatfunktioner.

ProduktID	Artnr	Produktnamn	Pris	typID
1	3443	Stolen Gunnar	123	12
2	5423	Bordet Jan	433	21
3	6534	Pallen Gun	199	12
4	6545	Kassen Jarl	12	2

Maxpris

433

Exempel:

```
SELECT MAX(pris) AS Maxpris
FROM Produkt
```



Exempel: COUNT räknar antal förekomster

```
SELECT COUNT(*) as AntaletKunder
FROM Kund
WHERE Ort='KALMAR';
```

AntaletKunder
1250

Exempel: SUM beräknar summa av värdet i angivet fält

```
SELECT SUM(Antal*pris) as Summan
FROM Artikel;
```

Summan
93750,50

Exempel: Avrundar nedåt.

```
SELECT FLOOR(SUM(Antal*pris)) as Summan
FROM Artikel;
```

Summan
93750

Exempel:

```
SELECT MAX(Antal), MIN(ANTAL), AVG(Antal)
FROM Fakturarad;
```

Max	Min	Avg
75	1	23



Vad är vår nettointäkt för varje rad på våra fakturor?

Syntax:

```
SELECT kolumn, Aggregatfunktion(kolumn)  
FROM tabell
```

Uppgift: Beräkna nettosumma för varje rad i fakturan.

Lösning:

```
SELECT SUM(fr.antal*fr.pris) AS Nettopris  
FROM Faktura AS f  
LEFT JOIN Fakturarad AS fr  
ON f.fakturaid = fr.fakturaid;
```

Nettopris
1250
1490
2100
1100
2870



Utgångspunkten är en eller flera tabeller där vi har flera poster som vi exempelvis vill summera och presentera ett slutresultat. En faktura innehåller flera rader och vi vill ha ett svar på hur är nettopriset för hela fakturan, dvs summan för alla raderna i fakturan:

Rader i Fakturarad

Fakturaid	Antal	Pris	Radsumma
1	10	125,00	1250
1	20	74,50	1490
Summa			2740
2	1	980,00	980
2	20	56,00	1120
Summa			2100
3	10	397,00	3970
Summa			3970

Det finns två rader i fakturarad som tillhör faktura 1

Dessa summeras till och presenteras som en rad genom GROUP BY i SQL-satsen:

Sammanställt med GROUP BY

Fakturaid	Datum	Nettopris
1	2008-02-15	2740
2	2008-03-21	2100
3	2008-04-12	3970



Med hjälp av GROUP BY kan vi se till att gruppera poster och på så sätt använda aggregatfunktioner på grupperna.

Syntax:

```
SELECT kolumn, Aggregatfunktion(kolumn)
FROM tabell
GROUP BY kolumn
```

Uppgift: Beräkna nettosumma för varje faktura.

Lösning:

```
SELECT f.Fakturaid, f.datum, SUM(fr.antal*fr.pris) AS Nettopris
FROM Faktura AS f
INNER JOIN Fakturarad AS fr
  ON f.fakturaid = fr.fakturaid
GROUP BY f.Fakturaid, f.datum;
```

Fakturaid	Datum	Nettopris
1	2008-02-15	2740
2	2008-03-21	2100
3	2008-04-12	3970



De fält i urvalet som inte omfattas av aggregatfunktion måste alla vara med i GROUP BY.

Exemplet från föregående sida:

Lösning:

```
SELECT f.Fakturaid,f.datum, SUM(fr.antal*fr.pris) AS Nettopris
FROM Faktura AS f
INNER JOIN Fakturarad AS fr
ON f.fakturaid = fr.fakturaid
GROUP BY f.Fakturaid, f.datum;
```

Fakturaid	Datum	Nettopris
1	2008-02-15	2740
2	2008-03-21	2100
3	2008-04-12	3970

Datum är omfattas av aggregatfunktionen MAX.

Alternativ lösning:

```
SELECT f.Fakturaid, MAX(f.datum), SUM(fr.antal*fr.pris) AS
Nettopris
FROM Faktura AS f
INNER JOIN Fakturarad AS fr
ON f.fakturaid = fr.fakturaid
GROUP BY f.Fakturaid;
```



HAVING

Med hjälp av HAVING kan vi utföra villkorsoperationer tillsammans med aggregatfunktionerna.

Syntax:

```
SELECT kolumn, Aggregatfunktion(kolumn)
FROM tabell
GROUP BY kolumn
HAVING Aggregatfunktion(kolumn) operator villkor
```

Uppgift: Beräkna nettosumma för varje faktura där $\text{sum} * \text{antal} > 2200$.

Lösning:

```
SELECT f.Fakturaid, f, datum, SUM(fr.antal*fr.pris) AS Nettopris
FROM Faktura AS f
INNER JOIN Fakturarad AS fr
ON f.fakturaid = fr.fakturaid
GROUP BY f. Fakturaid, f.datum
HAVING sum(fr.antal*fr.pris) >2200;
```

Om du vill ha ett urval på de poster som hämtas från tabellerna kan du använda WHERE och då läggs den innan GROUP BY

Fakturaid	Datum	Nettopris
1	2008-02-15	2740
3	2008-04-12	3970

Du kan inte använda ett alias i HAVING. Se nettopris ovan.